Iso2flux Manual

Installation

Linux/Mac

- Run the following code on the terminal: sudo apt-get update && apt-get -y install git python-dev libglpk-dev python-pip libfreetype6-dev libfreetype6 libpng-dev pkg-config libxml2-dev libxslt1-dev zlib1g-dev python-tk
- Run the following code on the terminal: sudo pip install --upgrade pip
- Run the following code on the terminal: sudo pip install -e git+https://github.com/cfoguet/iso2flux.git@master#egg=iso2flux

Optionally you can also install CPLEX or GUROBI for python following the instructions specific for each program. They are both available under academic license.

Windows:

- Download the latest version of Iso2flux from https://github.com/cfoguet/iso2flux.git@master
- Install Microsoft Visual C++ compiler for Python (https://www.microsoft.com/en-us/download/details.aspx?id=44266)
- Install GLPK for windows https://sourceforge.net/projects/winglpk/
    - Follow the installation instructions provided
- Install Python 2.7.x (latest version) https://www.python.org/downloads/windows/
    - Optional: Add python to the system path
- Run the following command in the terminal: pip install –upgrade Pip
- Download the appropriate versions of libsbml, numpy and scipy for your system and python installation http://www.lfd.uci.edu/~gohlke/pythonlibs/#numpy and install them using PIP
- Run the following command in the terminal: pip install -e “dir”
    - where dir is the directory where setup.py and the iso2flux folder are located (typically iso2flux-master)

Optionally you can also install CPLEX or GUROBI for python following the instructions specific for each program. They are both available under academic license.

Iso2Flux GUI

This part of the manual covers the use of Iso2flux through the GUI

Starting Iso2Flux GUI

To start the GUI of Iso2Flux use the command run_iso2flux_gui.py.

Selecting the working directory

After starting the GUI you will be prompted to select the directory where Iso2Flux will work.

Load/Create Iso2Flux instance

After selecting the working directory, you will be asked to choose between creating a new Iso2flux instance and loading an existing model Iso2Flux instance. Loading an instance will only work if the “equations” folder and its content created for it is present in the working directory selected.

Creating a iso2flux instance:

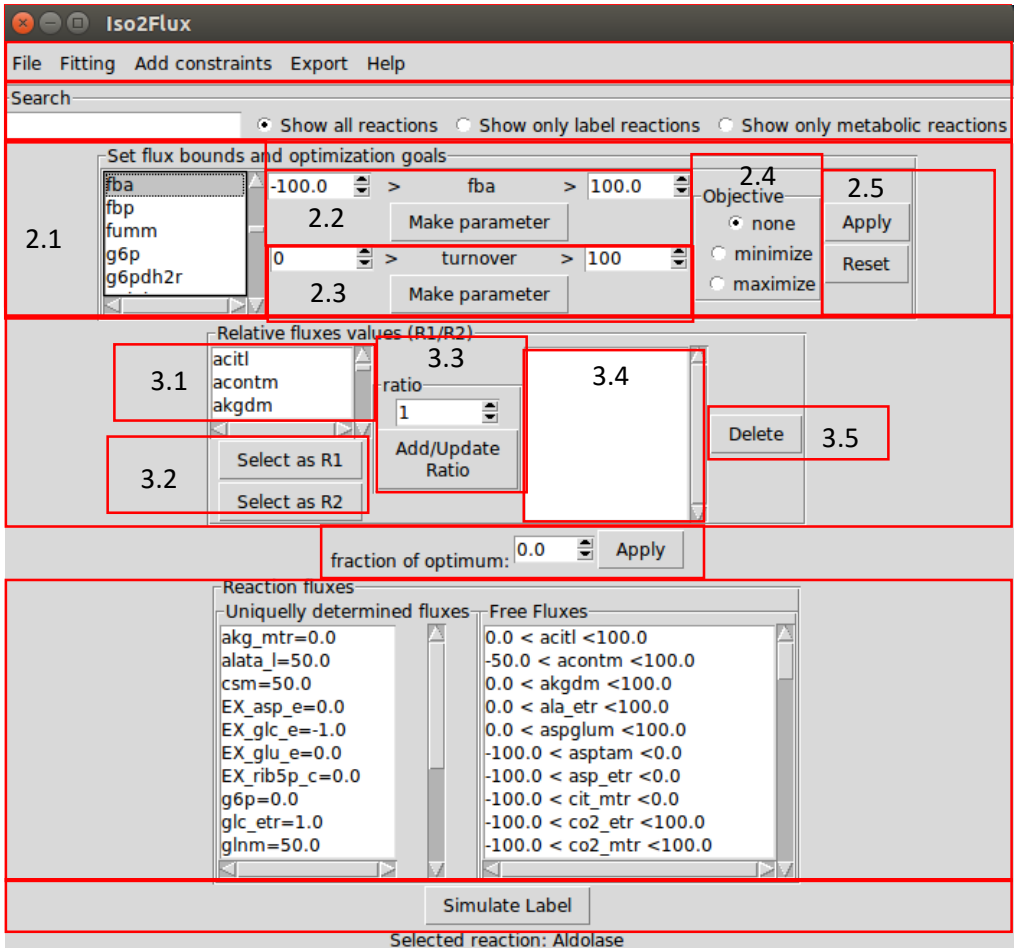
If you selected the option of creating a new instance you will have the defined the path to the multiple inputs taken by Iso2flux (See section inputs).

Validating the model

If it is the first time you are creating a model using a given COBA model and label propagation rules it is a good idea to check if the label propagation model is properly coupled to the the Constraint based model (i.e all metabolites in the label model are in steady state and no key reactions are missing). To do this click the Validate model option.

Main window

Once the model has been created or loaded the main iso2flux will open



0:Menubar

1:Reaction Filter

2:Set flux bounds, optimization goals and reaction turnover

3:Add or delete Ratios

4:Set Fraction of optimum

5:Flux distribution

6:Simulate Label

1: Reaction Filter

Allows to filter reactions based on reaction ID and reaction name by entering text into the search box. The radiobuttons can be used to either show all reactions, show only label reactions (reactions that directly propagate label) or only metabolic reactions (excluding reactions groups created by Iso2Flux.

2: Set flux bounds, optimization goals and reaction turnover

Setting the lower and upper bound of a reaction flux:

Select the reaction in the reaction selector (2.1) modify the upper and lower bound values in 2.2 and click apply (2.5) to commit the changes.

Setting a reaction flux to be maximized or minimized

Select the reaction in the selector (2.1), select the appropriate radiobutton in 2.4 and click apply (2.5) to commit the changes.

Setting the lower and upper bound of reaction turnover

If the selected reaction (2.1) directly propagates label and is reversible the reaction you can defined the upper and lower bound of the reversible turnover by changing the values (2.3) and committing the changes by clicking apply (2.5)

Manually setting a flux or a reversible turnover as parameter

To manually set a flux or a reversible turnover a parameter to be fit click the make parameter button in 2.2 and 2.3 respectively. However, generally is better to let Iso2Flux define the parameters through the automatically add parameters option in the fitting menu.

3 Add or modify Ratios

Adding a reaction ratio

To add a reaction flux ratio R1/R2=x, first select reaction R1 in the list (3.1), then click on "Select as R1" (3.2), then select reaction R2 in the list (3.1) and click on "Select as R2" (3.2). Finally introduce x in the entry box (3.3) and click on "Add/update ratio". If the ratio already exist it will be updated to the new value.

Removing an existing ratio:

To remove and existing ratio, select it on the list (3.4) and click on Delete (3.5)

4 Set Fraction of optimum

The fraction of optimum defines how much the flux distribution can diverge from the optimal solution. Setting the fraction to 1 will only select the optimal solution(s). If the objective is to maximize a flux the fraction of optimum can have any value between 0 and 1. If the objective is to minimize a flux the fraction of optimum can have any value larger than 1.

To change the fraction of optimum, enter the new value and click on apply

5: Flux distribution:

Displays the possible flux distribution obtained when solving the constraint based model with the currently defined constraints. The results are split into "Uniquely determined fluxes"(fluxes that can only have a possible value) and "Free fluxes" (fluxes that can have a range of values).

6: Simulating Label

Clicking on Simulate Label will open a window(s) comparing the experimentally determined isotopologues (bar chart) with those simulated by iso2flux with the current parameters and constraints (red dot). The window will be dynamically updated if constraints or parameters are modified.

0: Menubar:

File

Reset

Removes al changes (constraints, parameters etc..) made during the session

Save

Saves the current instance of iso2flux

Load:

Loads a previous instance of iso2flux

Fitting

Automatically add parameters

Automatically identifies the parameters that should be fitted. The user can choose to add either only flux value parameters, only reversible turnovers or both.

View parameters

Shows the currently active parameters and its value and offers a quick way of removing all active parameters.

Fit parameters

Finds the parameter values that allow to minimize the square deviation between experimentally measured isotopologues and those simulated by the model. It has the following parameters:

- N cycles: Number of simulated annealing cycles. At the end of each cycle the chance of accepting a worse solution will be decreased
- Simulations per cycle: Number of simulations at each cycle of simulated annealing
- P0: Probability of accepting a worse solution at the initial cycle of the simulated annealing
- PF: Probability of accepting a worse solution at the end of the last cycle of the simulated annealing
- Number of parallel process: Number of processes that the algorithm will use. Should not be larger than the number of CPU threads in the user computer
- Max size of parameter sample: Maximum number of parameters that will be changed at each annealing cycle. Cannot be larger than the total number of parameters
- Min size of parameter sample: Minimum number of parameter that will be changed at each simulated annealing cycle.

Sampling

Randomly samples the flux solution space and reversible turnover and performs a ^{13}C simulation at each step displaying the results in the label window(s). The user must choose the number of samples that will be taken. Sampling won't yield any result if the option to automatically add parameters has been used since there won't be any free fluxes affecting ^{13}C

Calculate confidence intervals

Used to compute the confidence intervals for fluxes after the parameters have been fit. The user must choose the significance level (e.g 95 for a 95% confidence level) and the step size used (i.e how much the evaluated parameters will increase at each reoptimization). Large step size can increase the speed of the analysis at the cost of reduced precision. The user must also choose the simulated annealing parameters (described above). If in addition of saving the results on a spreadsheet the user wants to save the confidence intervals as constraints in a SBML model the SBML checkbox can be ticked. Additionally, the user can tick the box "Add results as constraints" to use the results to constraint the currently active constraint based model.

Finally, the used can chose to start the analysis by clicking either the "Evaluate all fluxes" or "Evaluate specific fluxes & ratios". The first option will evaluate all parameters and provide confidence intervals for all fluxes in the model.

The second option will only be evaluating the fluxes and ratios defined in a file provided by the user. In the file the fluxes are defined by reaction ID and ratios are defined by having the ID of the two reactions involved separated by "/" (i.e R1_ID/R2_ID). Evaluate specific fluxes is faster than evaluate all fluxes provided the number of fluxes to evaluate is small.

Add constraints

Loads constraints

Adds constraints the defined in a file to the constraint based model. The format of the constraint file is defined in input section

Restrict max fluxes

Adds constraints to restrict the total flux amount in the model. The user must define tolerance of the constraints through the variable “Maximum flux/Min flux” which indicates how much the overall flux value is allowed to deviate from the minimum flux when setting the constraints. The user can also choose to load a file with metabolomics data to ensure that all metabolites that have been detected can be produced (see inputs for format).

Integrate gene expression

iMAT

Constraint the model integrating gene expression data using the GIM3E algorithm. It has the following parameters/inputs:

- Low expression percentile: Genes with expression level under this percentile will be considered lowly expressed.
- High expression percentile: Genes with expression levels over this percentile will be considered highly expressed.
- Epsilon: Minimum flux to consider a highly expressed reaction active.
- Fraction of optimal iMAT objective: Indicates how much can the solution deviate from the best solution found by the iMAT algorithm. Has range from 0 to 1 where 0 means it can deviate completely from the optimal iMAT solution and 1 means it cannot deviate from the optimal iMAT solution.
- Gene expression file: File containing a list of genes with its expression levels. See inputs for format.
- Metabolomics file: This input is optional. It ensures that all metabolites that have been detected can be produced after integrating gene expression data.

Note: To use iMAT either CPLEX or GUROBI extension for python must be installed, both of them are available under academic license.

GIM3E

Constraint the model integrating gene expression data using the GIM3E algorithm. It has the following parameters/inputs:

- Low expression percentile: Genes with expression level under this percentile will be considered lowly expressed.
- Fraction of optimal GIM3E objective: Indicates how much can the solution deviate from the best solution found by the GIM3E algorithm. Has range from 0 to 1 where 0 means it can deviate completely from the optimal GIM3E solution and 1 means it cannot deviate from the optimal GIM3E solution.
- Gene expression file: File containing a list of genes with its expression levels. See inputs for format.
- Metabolomics file: This input is optional. It ensures that all metabolites that have been detected can be produced after integrating gene expression data.

Export

Export Fluxes

Exports the flux distribution.

Export constraints

Export the constraints currently used in the constraint based model.

Export model

Exports the constraint based model with the currently defined constraints as either SBML model or as CSV or XLSX file.

Export label simulation results

Exports the results of the isotopologues simulation into a file.

Iso2flux command line script

This part of the manual covers the use of the command line script of Iso2flux *run_iso2flux.cl*. The script takes the following command line inputs (for details on the format of each file see Inputs Formats):

-e , --experimental_data_file= Name of file(s) containing the C13 patterns. If more than one file must be entered this can be defined by putting the name of the files in a string separated by a coma (e.g. "file1,file2")
-l , --label_propagation_rules= Name of the file describing the label propagation rules.
-c,--constrained_based_model= Name of the file (sbml, xslx or csv) describing the constraint based model that will be used

-f, --flux_constraints=: Name of the file describing additional constraints for the fluxes in the constraint based model
-p,--settings_file= Name of the file (xlsx or csv) defining additional settings for Iso2flux (Optional)
c-,--flux_constraints_file= Name of the file containing additional constraints for the constraint based model (Optional)
-o,--output_prefix= Prefix appended to all the output files (Optional)
-w,--working_directory= Name of the working directory (Optional). If none is defined it will use the one where the script is run
-g --gene_expression_file= Name of the file (xlsx or csv) indicating the gene expression in the conditions of study (Optional)
-m --metabolomics_file= Name of the file (xlsx or csv) indicating the metabolites that have been detected in the conditions of study (Optional). It will only be used if a gene expression file is provided.
-t --targetted_fluxes Name of the files (xlsx or csv) indicating the list of fluxes whose confidence intervals will be computed (Optional). If none is provided confidence intervals will be computed for all fluxes.

-q,--quick_analysis When this flag is used it disables the confidence interval analysis (Optional)

Once the script is initiated it will run from without requiring user input until the analysis is done. The following files will be generated:

- Initial_solution_space.csv: Indicates the possible flux distribution with the constraints defined in the CBM model and in the flux_constraint file
- Best_label.csv: Indicates the isotopologue fractions obtained in the best solution of the fitting algorithm
- Best_fluxes.csv: Indicates the flux distribution obtained in the best solution of the fitting algorithm
- Confidence.csv: Indicates the confidence interval for evaluated fluxes. Will only be generated if the flag -q is not set
- iMAT/Gim3e_solution.csv: Indicates the flux distribution resulting from integrating gene expression and metabolomics data. Will only be generated if a gene_expression file is defined in the input

Notes: If the user has added a "string" to the -o,--output parameter all the file names will be preceded by this string

Inputs formats

Constraint Based model file (Mandatory)

A CBM that provides the complete metabolic network stoichiometry, the default flux bounds and the gene-protein-reaction association used to integrate gene expression data. The CBM will be used to compute valid steady state flux distributions. The CBM can be entered either as a SBML file or as a CSV o XLSX file. An example of the information that should be included in CSV or XLSX file is provided bellow.

	A	B	C	D	E	F	G
1	**Reaction id**	**Stoichiometry**	**Reaction Name**	**Lower Bound**	**Upper Bound**	**Objective Coefficient**	**Gene rules**
2	glc_etr	glc_e <=> glc_c	Glucose extracellular transport	-1000	1000	0	gene1
3	hex1	atp_c + glc_c --> adp_c + glc6p_c	Hexokinase	0	1000	0	gene2 or gene3
4							
5	**Metabolite id**	**Metabolite Name**	**Formula**	**Compartment**			
6	glc_e	Glucose	C6H12O6	e			
7	glc_c	Glucose	C6H12O6	c			
8	atp_c	ATP	C10H16N5O13P3	c			
9	adp_c	ADP	C10H15N5O10P2	c			
10	glc6p_c	Glucose 6-pshosphate	C6H12O6	c			

A

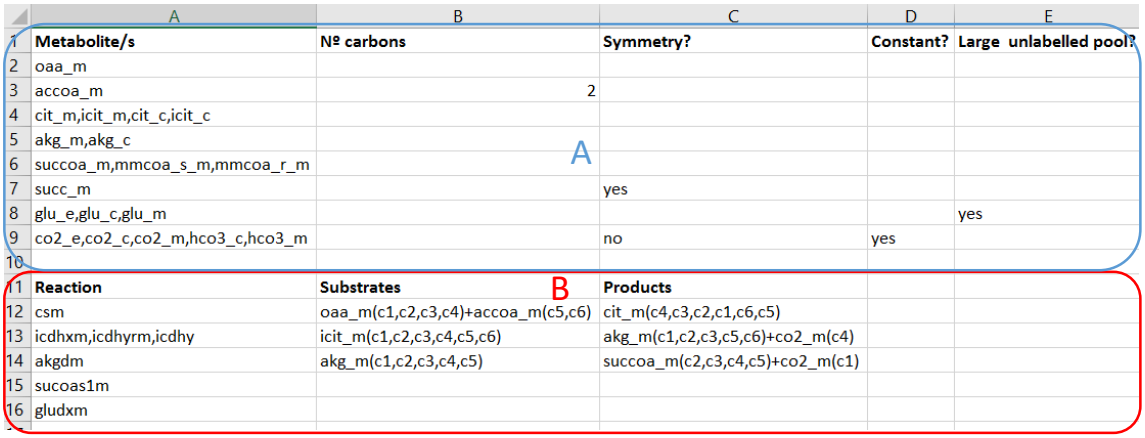
B

- A: Reactions (Mandatory): Describes the reactions of the CBM
    - Reaction ID (Mandatory): ID that will be given to the reaction. ID must be unique to a given reaction
    - Stoichiometry (Mandatory): Stoichiometry of the reaction
    - Reaction Name (Optional): Name of the reaction
    - Lower Bound (Optional): Minimal value of the flux through the reaction
    - Upper bound (Optional): Maximal value of the flux through the reaction
    - Objective coefficient (Optional): Weight given to the reaction flux to the objective function. If its positive the flux will be maximized if its negative the flux will be minimized and if its 0 it won't be part of the objective function
    - Gene rules: Indicates the genes associated to each reaction following the standard gene protein reaction (GPR) notation.
- B: Metabolites (Optional): Describes the metabolites in the CBM. If they are not defined they will be automatically identified from the reaction stoichiometry.
    - Metabolite ID: ID of the metabolite, must appear at least once in the stoichiometry of one the reactions defined.

- Metabolite Name: Name of the metabolite
- Metabolite Formula: Empirical formula of the metabolite
- Compartment: Cellular compartment where the metabolite is found

The label propagation rules (Mandatory):

A XLSX or CSV file that defines how label is propagated in the model. It contains 2 type of information, metabolites where label can propagate and its attributes and reactions that propagate label and its attributes. An example indicating the different options is provided bellow



	A	B	C	D	E
1	**Metabolite/s**	**Nº carbons**	**Symmetry?**	**Constant?**	**Large unlabelled pool?**
2	oaa_m				
3	accoa_m	2			
4	cit_m,icit_m,cit_c,icit_c				
5	akg_m,akg_c				
6	succoa_m,mmcoa_s_m,mmcoa_r_m				
7	succ_m		yes		
8	glu_e,glu_c,glu_m				yes
9	co2_e,co2_c,co2_m,hco3_c,hco3_m		no	yes	
10					
11	**Reaction**	**Substrates**	**Products**		
12	csm	oaa_m(c1,c2,c3,c4)+accoa_m(c5,c6)	cit_m(c4,c3,c2,c1,c6,c5)		
13	icdhxm,icdhyrm,icdhy	icit_m(c1,c2,c3,c4,c5,c6)	akg_m(c1,c2,c3,c5,c6)+co2_m(c4)		
14	akgdm	akg_m(c1,c2,c3,c4,c5)	succoa_m(c2,c3,c4,c5)+co2_m(c1)		
15	sucoas1m				
16	gludxm				

A Metabolites where label is propagated: Contains the following information:

- Metabolite/s: ID(s) of the metabolites that can propagate label. Should be the same
- metabolite IDs used in the CBM. When several IDs are entered separated by a coma it indicates that these metabolites should be grouped into a single label pool
- Nº Carbons: Number of positions that can contain labelled atoms. If left empty it will automatically get the number of carbons from the empirical formula defined in the CBM. In the example above, the field has been filled for acetyl CoA (accoa_m) because the number of carbon atoms in the formula (23) is different from the number of carbons when we assume label can propagate (2)
- Symmetry: Use Yes to indicate that the metabolite/s are symmetric like it is the case with succinate. If left empty it will assume the molecule is not symmetric
- Constant: Use Yes to indicate that the abundances of isotopologues for these metabolite should be constant. Constant isotopologues will always have their initial abundances. In the example above we define that the fractions of isotopologues for carbon dioxide and carbonate pool will be constant and, unless they are defined as labelled in the experimental data file, it always be a 100% m0. If left empty it will be assumed the the isotopologues for this metabolite are not constant.

B Reactions that can propagate label. Contains the following information:

- Reaction ID(s): ID of the reactions that can propagate label. Should be the same reaction IDs used in the CBM. If several reactions IDs are listed separated by coma it will group, the reactions as described in methods. In the example above we wish to group all the

isocitrate dehydrogenase enzymes (icdhxm, icdhyrm and icdhy) because from a label propagation perspective they all are equivalent

- Substrates and Products: This should indicate how label is propagating by the reaction. For substrates the substrates ID should be written followed by the identification given to the atoms of the substrate that can be labelled (in order). Products should indicate the ID of the products of the reaction followed by the identification of the atoms of the substrates that form the product (in order). Any identification for atoms can be used provide they are consistent between substrates and products and they are not repeated within the same reaction. When the substrates and products are part of a metabolite group the ID of any member of the group can be used. If the field substrates and products are left empty it will be assumed that the reaction does not alter the position of the labelled atoms between the substrate and product like is the case of glutamate dehydrogenase (gludxm) in the example above. Input and Output reactions do not need to be defined since they will be automatically added by Iso2Flux.

Iso2Flux is programed to recognize whether a given row contains information about metabolites (A) or reactions (B) or neither.

^{13}C patterns files (mandatory)

One or more XLSX or CSV file that defines the labelled substrates used and the quantified isotopologues. Below is an example of the information that the file should contain.

	A	B	C	D	E	F	G
1	**Labeled_substrates**						
2	Metabolite id	Label pattern	Abundance	A			
3	glc_e	1,1,0,0,0,0	0.5				
4							
5	**Isotopologue distribution in products**						
6	Name	Metabolite id	Positions	Isotopologue	Mean	SD	compute m/Sm?
7	Lac-328	lac_e	1-3	m0	0.80	0.00	
8				m1	0.00	0.00	
9				m2	0.19	0.00	
10				m3 B	0.00	0.00	
11							
12							
13	Rib-256	atp_c	1-5	m0	0.84	0.02	yes
14				m1	0.05	0.00	
15				m2	0.08	0.01	
16				m3	0.01	0.00	
17				m4	0.02	0.00	
18				m5	0.00	0.00	
19							
20	Glu-198	glu_e	2-5	m0	0.99	0.00	
21				m1	0.00	0.00	
22				m2	0.01	0.00	
23				m3	0.00	0.00	
24				m4	0.00	0.00	

A Labelled substrates used and its abundance. Contains the following information:

- Metabolite ID, ID of the labelled substrates, must be the same used in the CBM
- Label pattern, indicates of the positions of the metabolite that contain a heavy isotope. 1 denotes heavy isotope (e.g. ^{13}C) and 0 denotes non-heavy isotope (^{12}C). In the example above 1,1,0,0,0,0 indicates we have glucose labelled with ^{13}C in the first two positions.
- Abundance, indicates the relative abundance of the labelled substrate (e.g. 0.5 indicates 50% abundance).

It is not necessary to indicate unlabelled substrates (e.g. in the example above is not necessary to indicate that the abundance of the unlabelled substrate is 0.5)

B Isotopologue distribution in products, contains information about the quantified isotopologues and is abundance. Specifically, the following information:

- Name of the measurements. This will be used in the output and graphical interface to refer the measurements.
- Metabolite id. Id of the metabolite has been measured, must be the ID used in the CBM. If the measurement correspond to multiple metabolites they should be grouped in the label propagation rules and any of the metabolites ID grouped indicated in this field.
- Positions measured: Starting and end position of the fragment that has been measured. In the case of Glu-198, 2-5 indicates that we have measured isotopologues in the a fragment that contains only carbons 2 to 5.
- Isotopologue: Indicates the number of heavy isotopes present in each isotopologue (i.e m0:0, m1:1, m2:2 etc …)
- Mean: Arithmetic mean of the relative abundance of each isotopologues in the replicates.
- SD: Standard deviation of the measurements of relative abundance in a isotopologue
- m/Sm: Use Yes to indicate that you wish Iso2Flux to work with the relative abundance of labelled isotopologues. As described in methods this is one of the ways of working with metabolites with large unlabelled pools. In the example above we use this option with the label measured on nucleotides because there can be a significant pool of unlabelled nucleotides originating from the initial RNA.

The fields Name of the measurements, Metabolite id, Positions measured and m/SM only need to be entered in the first row of each quantified fragment.

Constraints file (optional):

A XLSX or CSV file that defines constraints for the CBM. In addition to defining reactions and bounds and objective coefficients can also be used to define flux ratios and group reactions. An example of each input is provided bellow.

	A	B	C	D	E
1	**Reaction id**	**Lower bound**	**Upper bound**	**Objective coefficient**	A
2	EX_ala_e	-0.007138141	0	0	
3	**Reaction Groups**				B
4	EX_glu_e+2*EX_gln_e	-2000	1000	0	
5	**Ratios**	Ratio value			C
6	g6pdhr2/hex1	0.01			

A: When a reaction ID is found on the first column the second column will define the lower bound of the reaction, the third column the upper bound of the reaction and the fourth column the objective coefficient (if its positive the flux will be maximized and if its negative the flux will be minimized).

B: When several reaction IDs are found in the first column separated by a "+" a group reaction will be created for these reactions. The lower bound and upper bounds and objective coefficient of the group reaction can be defined like in A

C: When 2 reaction IDs separated by "/" are found in the first column the flux of the two reactions will be forced to have the ratio defined in the second column.

Gene expression file (optional):

A XLSX or CSV file that defines the gene expression. In the first row it should have the gene identifiers and the second column the gene expression value. It is important that the type of identifier used in the file is the same as the one used in the CBM model.

Metabolomics file (optional)

A XLSX or CSV file that indicates the metabolites that have been detected in the study conditions. Metabolites can be indicated either from metabolite name or metabolite ID.

Targeted fluxes (optional)

A XLSX or CSV file that indicates specific fluxes or ratios for which the confidence intervals should be computed. In the file the fluxes are defined by reaction ID and ratios are defined by having the ID of the two reactions involved separated by "/" (i.e R1_ID/R2_ID). If this file is not provided all fluxes will be evaluated.

Step by step guide to computing confidence intervals

Using the GUI:

It is expected the most common use of Iso2Flux will be to compute confidence intervals for all fluxes of the model given a set of experimental data. To do this the following steps can be followed.

- Start the GUI by executing "python run_iso2flux_gui.py"
- Load or create the Iso2flux instance following the instructions defined in the section *Load/Create Iso2Flux instance*
- Add constraints based on experimentally measured fluxes **(Optional)**: Modifying the lower and upper bound the fluxes in the *Set flux bounds, optimization goals and reaction turnover* panel to be in accordance to experimental measurements. Alternatively write a file with the constraints and load them through the "Load constraints" option in the "Add constraint" menu.
- Automatically select parameters to fit: Go to Fitting/Automatically add parameters and make sure both options are checked
- Find the parameter set that minimized the differences between experimental and simulated isotopologues: Go to Fitting/Fit parameter and click start. Optionally you can tweak the annealing parameters to adjust the behaviour of the fitting algorithm.
- Computing the confidence intervals: Go to Fitting/Confidence intervals and select Evaluate all fluxes, chose where do you want to save the output and wait for the analysis to be done. Optionally check the option "Add results as constraints" to use the results to constraint the model.
- Integrate gene expression data **(Optional)**:
    - Requires having used the option "Add results as constraints" in the confidence intervals analysis
    - Use Add Constraint/Integrate gene expression/iMAT or Add Constraint/Integrate gene expression/GIM3E to constraint the confidence intervals result with gene expression data

- Use Export/Export fluxes to save the resulting flux distribution

Specific instructions on how to use the described function is provided elsewhere in the manual. In most circumstances the default value of parameters should be adequate.

Using the command line script:

Execute the following on the terminal python *run_iso2flux_cl -e 13cpatterns.csv -l label_propagation_rules.csv -c constrained_based_model.sbml -f flux_constraints.csv*

Where:

13cpatterns.csv : is a ^{13}C pattern file

label_propagation_rules.csv: is a file describing the label propagation rules

c constrained_based_model.sbml : Is the constrained based model

f flux_constraints.csv: Is a constraints file describing the lower and upper bound of the fluxes measured experimentally

Optionally, add the following parameters -g gene_expression.csv and -m metabolomics.csv to integrate gene expression (gene_expression.csv) and metabolomics data (metabolomics.csv)

The program will run without requiring further user input and the confidence intervals for fluxes will be saved in a file named “confidence.csv”